Listing of Claims:

1. (Previously Presented)   A system for enabling socket communication in COBOL program, comprising:

>   a memory block;

>   a COBOL program communicating with the memory block;

>   a socket; and

>   a COBOL routine callable from the COBOL program, the COBOL routine reads information from the socket and writes the information read from the socket to the memory block in response to the COBOL program call, wherein the COBOL routine reads the information from the socket through a bit-level call to an operating system.

2. (Original)   The system of Claim 1, wherein the COBOL program further communicates with the COBOL routine to initiate the COBOL routine communication with the socket and the memory block.

3. (Original)   The system of Claim 1, wherein the COBOL routine is further defined as a subroutine of the COBOL program.

4. (Original)   The system of Claim 1, wherein the COBOL routine is further defined as a COBOL library having a plurality of routines callable by the COBOL program.

5. (Original)   The system of Claim 1, wherein the COBOL routine is further defined as a compiler enabled function usable by the COBOL program.

2

6. (Previously Presented) A method for enabling socket communication in COBOL program, comprising:

> requesting, by a COBOL program stored on a computer readable medium, information from a socket;

> retrieving, by a COBOL routine stored on a computer readable medium, information from the socket through a bit-level call to an operating system;

> writing, by the COBOL routine, information read from the socket to a memory block; and

> reading from the memory block, by the COBOL program, the information.

7. (Original) The method of Claim 6, wherein the method further comprises managing, by the COBOL routine, a connection with the socket.

8. (Original) The method of Claim 7, wherein managing includes listening on the socket connection.

9. (Original) The method of Claim 7, wherein managing includes disconnecting the connection with the socket.

10. (Original) The method of Claim 6, wherein the method further comprises, establishing, by the COBOL routine, a connection with the socket.

11. (Previously Presented) The method of Claim 10, wherein the connection with the socket is established in response to a request from the COBOL program.

3

12. (Original) The method of Claim 6, wherein the COBOL routine provides an address to the COBOL program, the address identifying a location of the memory block where the information is written.

13. (Original) The method of Claim 12, wherein the method further comprises mapping, by the COBOL program, the memory block into the COBOL program.

14. (Original) The method of Claim 13, wherein the mapping is accomplished using a copybook.

15. (Original) The method of Claim 6, wherein the information is provided in an EBCDIC format and wherein the method further comprises converting the information from the EBCDIC format to an ASCII format.

16. (Original) The method of Claim 15, wherein the conversion is accomplished by the COBOL routine.

17. (Original) The method of Claim 6, wherein the COBOL routine further includes a coordination module to coordinate such that the COBOL routine only reads when the socket has information and only writes when the socket is not full.

18. (Original) The method of Claim 6, further comprising initiating a call to the operating system by the COBOL routine to establish a socket connection.

4

19. (Previously Presented) The method of Claim 18, wherein the call to the operating

    system is further defined as a bit-level call to the operating system of a mainframe

    computer system.

20. (Original) The method of Claim 19, wherein the COBOL routine is further defined as

    written in COBOL programming language.

5

21. (Previously Presented) A system for enabling pipe communication in COBOL programs, comprising:

a memory block;

a COBOL program stored on a computer readable medium communicating with the memory block;

a pipe; and

a COBOL routine stored on a computer readable medium callable from the COBOL program, the COBOL routine reads information from the pipe and writes the information read from the pipe to the memory block in response to the COBOL program call, wherein the COBOL routine reads the information from the pipe through a bit-level call to an operating system.

22. (Previously Presented) The system of Claim 21, wherein the memory block is further defined as a mainframe memory block and wherein the COBOL program and the COBOL routine are operable on a mainframe computer system.

23. (Previously Presented) The system of Claim 22, wherein the COBOL routine is further defined as a COBOL technical layer having a plurality of routines callable by the COBOL program, the COBOL technical layer including:

a create module communicating with a computer system to create a pipe connection;

a connect module that promotes attachment to the pipe connection;

6

an open module that opens the pipe connection to promote communication via

the pipe connection;

a write module that writes information to the pipe connection, the write module

verifies that the pipe connection is not full prior to writing information and

blocks when the pipe connection is full;

a read module coupleable to the pipe connection to read information from the

pipe connection;

a release module to release the pipe connection;

a remove module to remove the pipe connection from the computer system;

and

a delete module to delete the pipe connection wherein the pipe connection is

closed.

7

24. (Previously Presented) A method for socket communication in COBOL, comprising:

reading, by a routine stored on a computer readable medium, information from

a socket through a bit-level call to an operating system;

writing, by the routine, the information to an area; and

reading, by a COBOL program stored on a computer readable medium, the

information from the area, the COBOL program and the routine operating

in the same runtime environment.

25. (Original) The method of Claim 24, wherein the area is a file.

26. (Original) The method of Claim 24, wherein the area is a memory area.